CS 2300 Module 1 Open-Ended Project

For this project, you will design, implement, and use a C++ project.

Requirements

- All of your programming files should be in a **private** GitHub repository in the course organization.
  - **Your repository must be named with the convention: M1OEP-netid**, where netid is your UVM NetID username.
  - The repository should have a .gitignore file, a CMakeLists.txt file, and a README file (see the GitHub-with-CLion repo for directions).
- Your README.md file should include:
  - Your name
  - A summary of your program, including a list of Module 1 concepts used
  - Any known bugs at time of submission
  - Future work (how you could expand the program with more time)
  - Citations for any code not written by you or the instructor
  - The grade you think you have earned, based on the grading rubric below, with justification
- All of your files must be pushed to your GitHub repository.
- Create a short video (~5 minutes) giving a brief overview of the program, including a demo.
  - You can record the video on Microsoft Teams or a different program of your choosing. Instructions to record in Teams are at the bottom of this spec.
- Submit both the link to your repository and the link to your video.
  - You can upload your video to OneDrive or a different platform of your choosing to obtain a link.

Main Program

Your main program should have a clear and complex purpose. It should **not** be used as a testing file to call methods and functions. The goals and functionality of your program should inform the development and design of your class(es) and other concepts. You are highly encouraged to plan and design your main.cpp functionality before writing your other files.

The program should be interactive and all user input must be validated.

Keep in mind that you will be allowed and encouraged to build on this project for future modules' open-ended projects, so putting the time in now to construct a strong foundation will be to your advantage.

Module 1 Concept Menu

You can include any number and combination of these C++ concepts from Module 1:

- Class (separated into header and .cpp files, containing at least two fields and at least one method that isn't a constructor, getter, setter, or destructor)
- Overloaded Operator(s) in your class(es)
- Enumerated type(s)

All of the above concepts that you design, create, and use in your main program must be clearly listed in your README.md file.

Grading

The size and functionality of your program is the single item worth the most points, so make sure you design well before you start coding.

	N/A (0 pts)	Poor (5 pts)	Good (20 pts)	Great (40 pts)
Main program complexity and usability	No main function or little/ no functionality in main function or doesn't run	Testing program or no user interaction	Main program uses some concepts from Module 1; interactive; as complex as a guided project	Main program uses many concepts from Module 1; interactive with all user input validated correctly; sufficiently complex for an open-ended project

The more concepts from the module you include, the more points you can get! **You can include more than one of each concept, and each will be graded separately.** Note that if two or more of a concept are deemed too similar, you cannot get full points for each. Choose wisely.

	N/A (0 pts)	Poor (5 pts)	Good (10 pts)	Great (20 pts)
Concept: C++ class	No C++ class present in the program	Class is trivial, poorly designed, and/ or poorly implemented	Class has some design/ implementation errors but is used in the main program	Class is designed well, implemented robustly, and used in the main program in a way that makes sense

	N/A (0 pts)	Poor (2 pts)	Good (5 pts)
Concepts: overloaded operator, enum	Not present in the program	Concept is trivial, poorly designed, and/ or poorly implemented	Concept is designed well, implemented robustly, and used in the main program in a way that makes sense

The rest of the items are expected; failure to complete them will result in a loss of points.

	N/A (-20 pts)	Poor (-10 pts)	Default (0 pts)
Style and documentation	Not present in the program/repository	Some comments and documentation, some organization	Robust comments and documentation, clearly organized. Classes are separated into .h and .cpp files.

	N/A (-10 pts)	Poor (-5 pts)	Fair (-2 pts)	Good (0 pts)
Video	No video or video permissions aren't set to be visible to graders	Video has no demo run	Video includes at least one demo run but does not demonstrate input validation	Video includes multiple demo runs and demonstrates input validation

	: ((-10 pts)	: / (-5 pts)	:) (0 pts)
Lifespan of project	Repository contributions are all made within 72 hours	Repository commits/contributions are spread over more than 3 days	Repository commits span at least 7 days and significant contributions are made on at least 3 separate days

If you do absolutely everything you can, you can earn over 100 points. Realistically, be happy with 60 points (which is still more than any other project option in this course!).

How to Create a Video in MS Teams

1) Go to Calendar.
2) Click the "Meet now" button.
3) The name of the meeting defaults to "Meeting with [your name]" but you can edit it. Click the "Start meeting" button.
4) Now you have options:
   - Make sure your audio is on.
   - It's nice to also have video, if it's available to you.
   - There may also be the option to blur your background or choose a background image. Here's a chance for creativity.
5) Click the "Join now" button.
6) Out of the options across the top of the window, click on the three dots icon for more options. This is where you will find the button to "Start recording".
7) Click on the "Share content" button (to the right of the audio and video buttons). Choose your share option.
   - This will make the Teams window minimize, and you will see a small black box in the corner of your screen. The part of your screen that is being shared is outlined in red.
8) Once you are finished the demo, click the red hang-up button.
9) Go to the Chat tab on Teams. There should be a chat there with the name of your meeting. Click on it.
10) When your recording is ready (it doesn't usually take long), a thumbnail will appear in the chat. Click the three dots in the upper-right corner of the video and choose "Open in Stream".
11) This will open the video in OneDrive/Sharepoint. Click the Share button in the top right corner, then click "Copy link".
12) Click Settings and then toggle off the "Block download" option. Then click Apply. It should now read "People in your organization with the link can view."
13) Paste the link in the Gradescope submission box for this project.